# Final Project

2025-07-12

```
library(readr)
proj_genes <- read_csv("/Users/ayei/Desktop/QBS103_GSE157103_genes.csv")
```

```
## New names:
## Rows: 100 Columns: 127
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): ...1 dbl (126): COVID_01_39y_male_NonICU, COVID_02_63y_male_NonICU,
## COVID_03_33y_...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

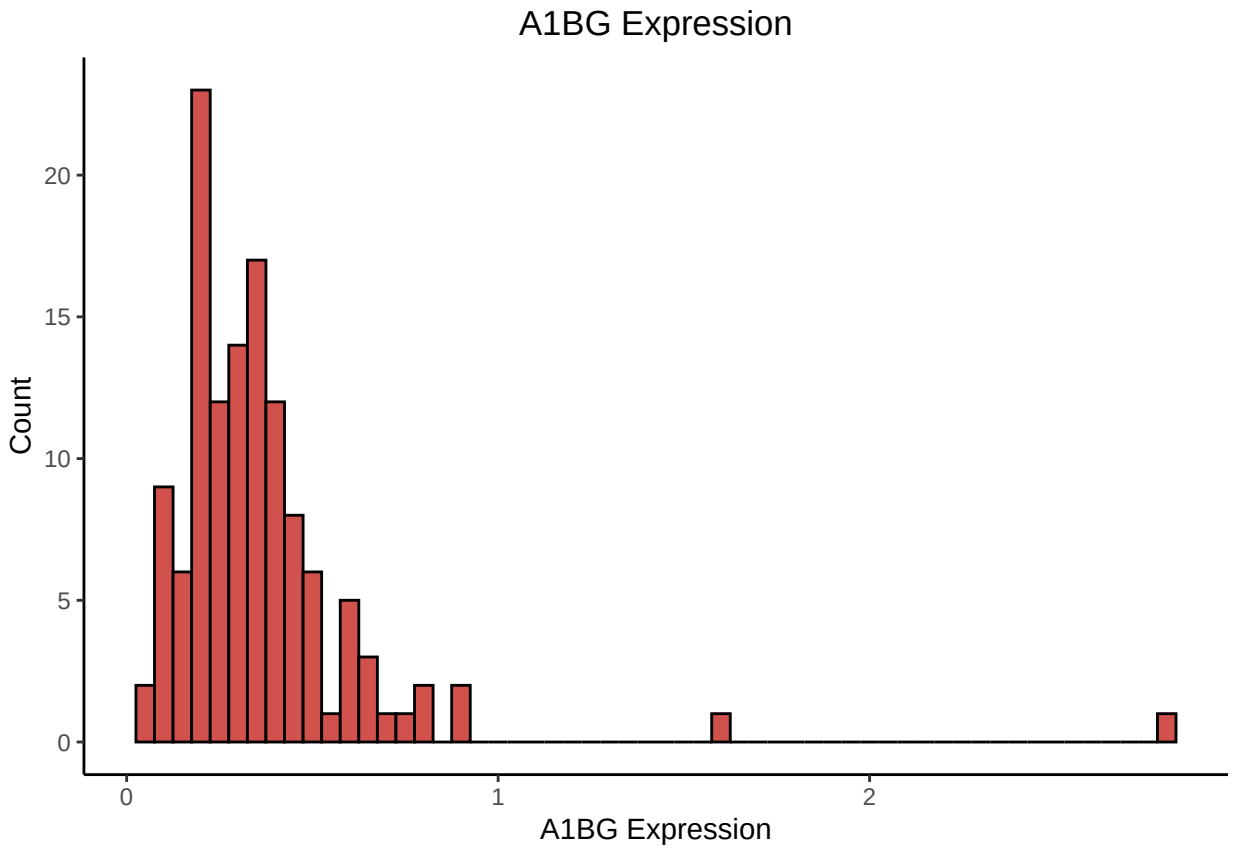
```
library(readr)
proj_matrix <- read_csv("/Users/ayei/Desktop/QBS103_GSE157103_series_matrix-1.csv")
```

```
## Rows: 126 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (21): participant_id, geo_accession, status, !Sample_submission_date, la...
## dbl  (4): channel_count, charlson_score, ventilator-free_days, hospital-free...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Because I need to match the participant_id column/row position in order to link the two data sets, I

# Converting raw data to data frame
proj_genes_df <- as.data.frame(proj_genes)

# Setting the row names as the gene names
rownames(proj_genes_df) <- proj_genes_df[[1]]

# Now I have to remove the first column since I am using them as row names
proj_genes_df <- proj_genes_df[ , -1]

# Transposing the data so that the gene names go across the top and the participant ID's became the col
new_proj_genes <- as.data.frame(t(proj_genes_df))

# Just doudble checking
# colnames(proj_matrix)
# colnames(new_proj_genes)

# Now I am adding a new column for "participant_id" to "new_proj_genes" with the row names, so that whe
new_proj_genes$participant_id <- rownames(new_proj_genes)
```

```r
# Merging the two datasets
merged_proj_data <- merge(proj_matrix, new_proj_genes, by = "participant_id")

# Double checking to see if the merge was successful
# summary(merged_proj_data)
```

```r
library(ggplot2)

# Making my histogram for A1BG expression
ggplot(merged_proj_data, aes(x = A1BG)) +
  # Decided to make my binwidth 0.05 for more data points/make the data points more distinct
  # Also customized my colors
  geom_histogram(binwidth = 0.05, fill = "#D1514D", color = "black") +
  # Added a title, and x and y axis labels
  labs(title = "A1BG Expression",
       x = "A1BG Expression", y = "Count") +
  # Set the theme to classic for a clean background
  theme_classic() +
  # Wanted to center my title
     ##https://www.r-bloggers.com/2025/03/how-to-center-ggplot-title-subtitle-and-caption-in-ggplot2-wi
  theme(plot.title = element_text(hjust = 0.5))
```

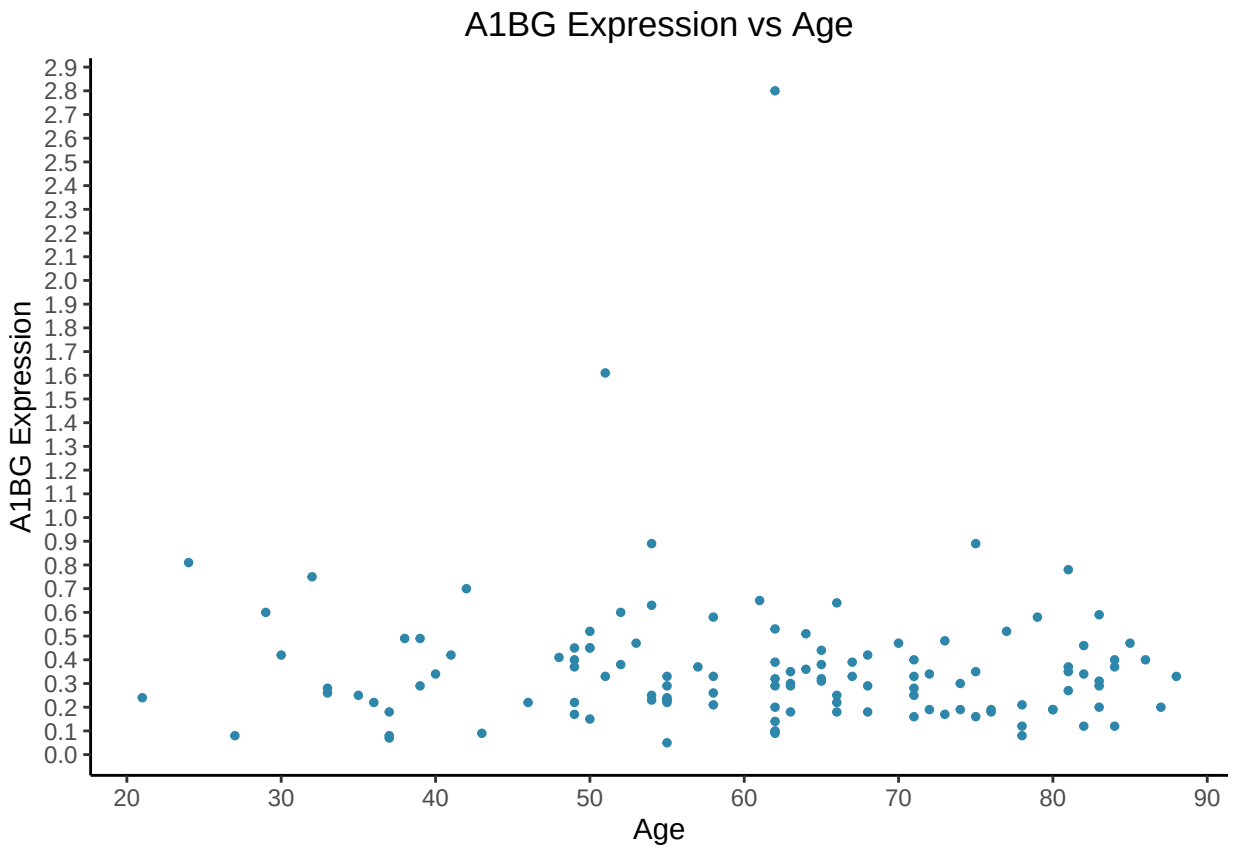



```r
# "age" was not being registered as a number, so I had to change the age column to numeric
merged_proj_data$age <- as.numeric(merged_proj_data$age)
```

```
## Warning: NAs introduced by coercion
```

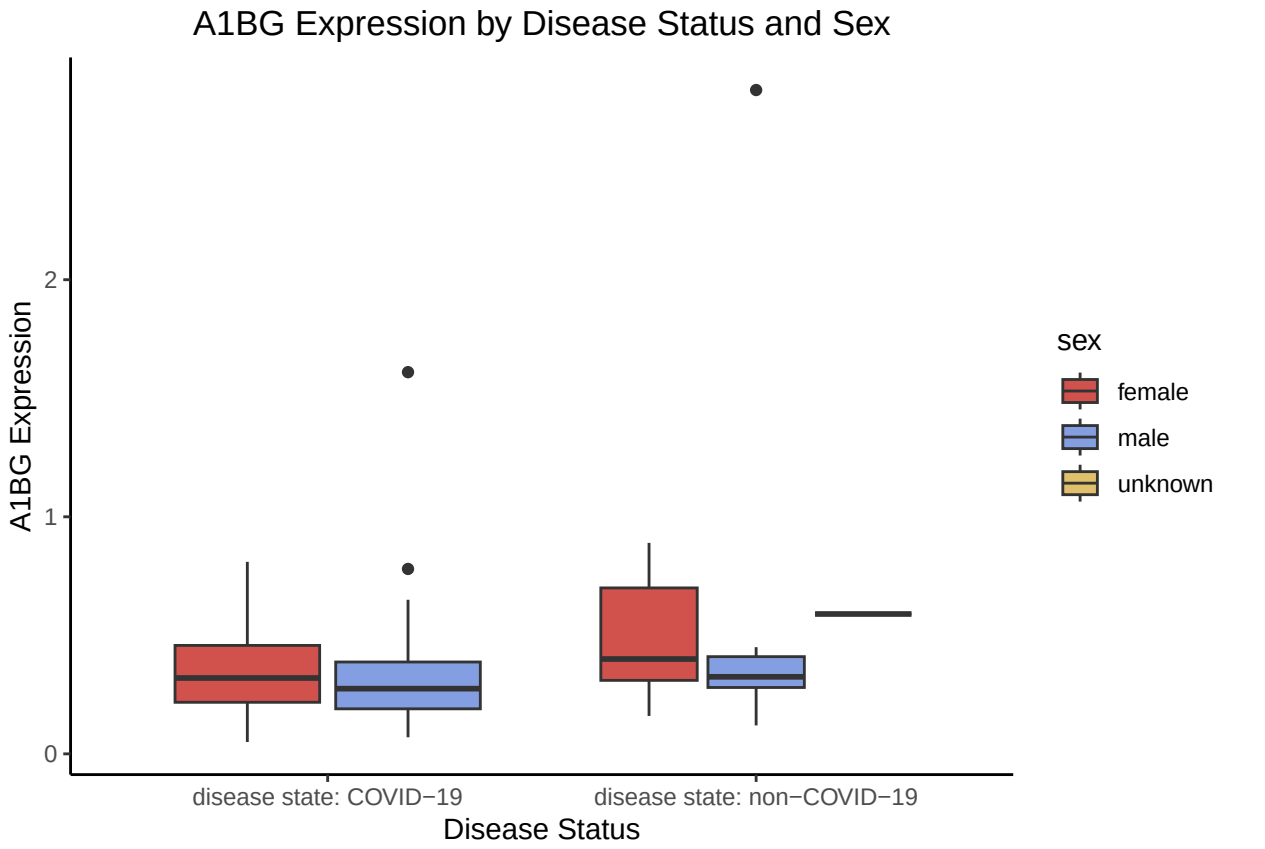
```r
# Making my scatterplot for A1BG Expression vs Age
ggplot(merged_proj_data, aes(x = age, y = A1BG)) +
  # Customized the color and the size -- I set the size to 1 so that the point is more precise (when it
  geom_point(color = "#2E86AB", size = 1) +
  # Added a title, and x and y axis labels
  labs(title = "A1BG Expression vs Age",
       x = "Age", y = "A1BG Expression") +
  # Customized the scale for x axis to go up by 10s to make the plot easier to read
      ##https://www.sthda.com/english/wiki/ggplot2-axis-scales-and-transformations#google_vignette
  scale_x_continuous(breaks = seq(0, 100, by = 10)) +
  # Customized the scale for y axis to go up by 0.1 to make the plot easier to read
  scale_y_continuous(breaks = seq(0, 3, by = 0.1)) +
  # Set my theme to classic for a clean background
  theme_classic() +
  # Centered the title
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```r
# Making my boxplots for A1BG expression by disease status and sex
ggplot(merged_proj_data, aes(x = disease_status, y = A1BG, fill = sex)) +
  geom_boxplot() +
  # Customized the colors for sex
  scale_fill_manual(values = c("male"= '#849FE1', "female" = '#D1514D',"unknown" = '#E0C06B')) +
  # Added a title, and x and y labels
  labs(title = "A1BG Expression by Disease Status and Sex",
```

```
      x = "Disease Status", y = "A1BG Expression") +
  # Set my theme to classic for a clean background
  theme_classic() +
  # Centered my title
  theme(plot.title = element_text(hjust = 0.5))
```

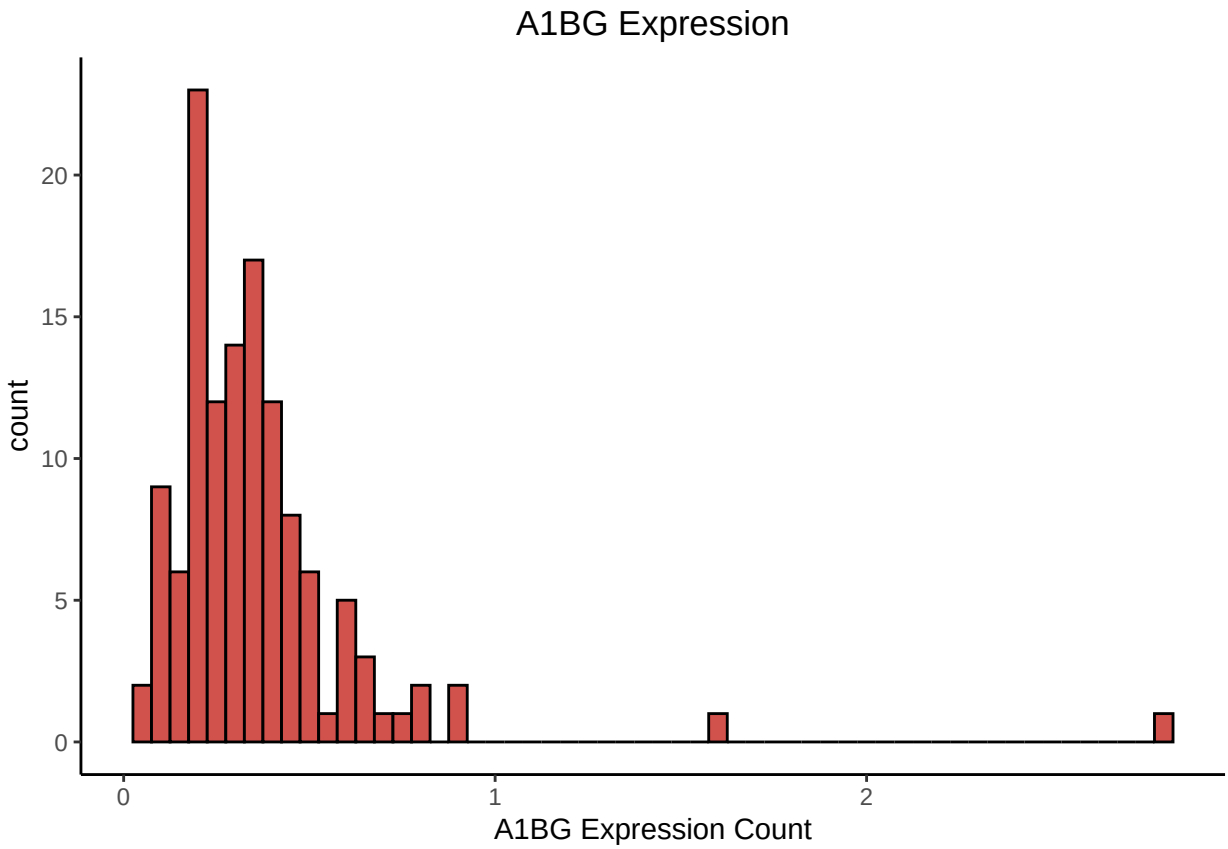



Build a function to create the plots you made for Presentation 1, incorporating any feedback you received on your submission. Your functions should take the following input: (1) the name of the data frame, (2) a list of 1 or more gene names, (3) 1 continuous covariate, and (4) two categorical covariates (10 pts)

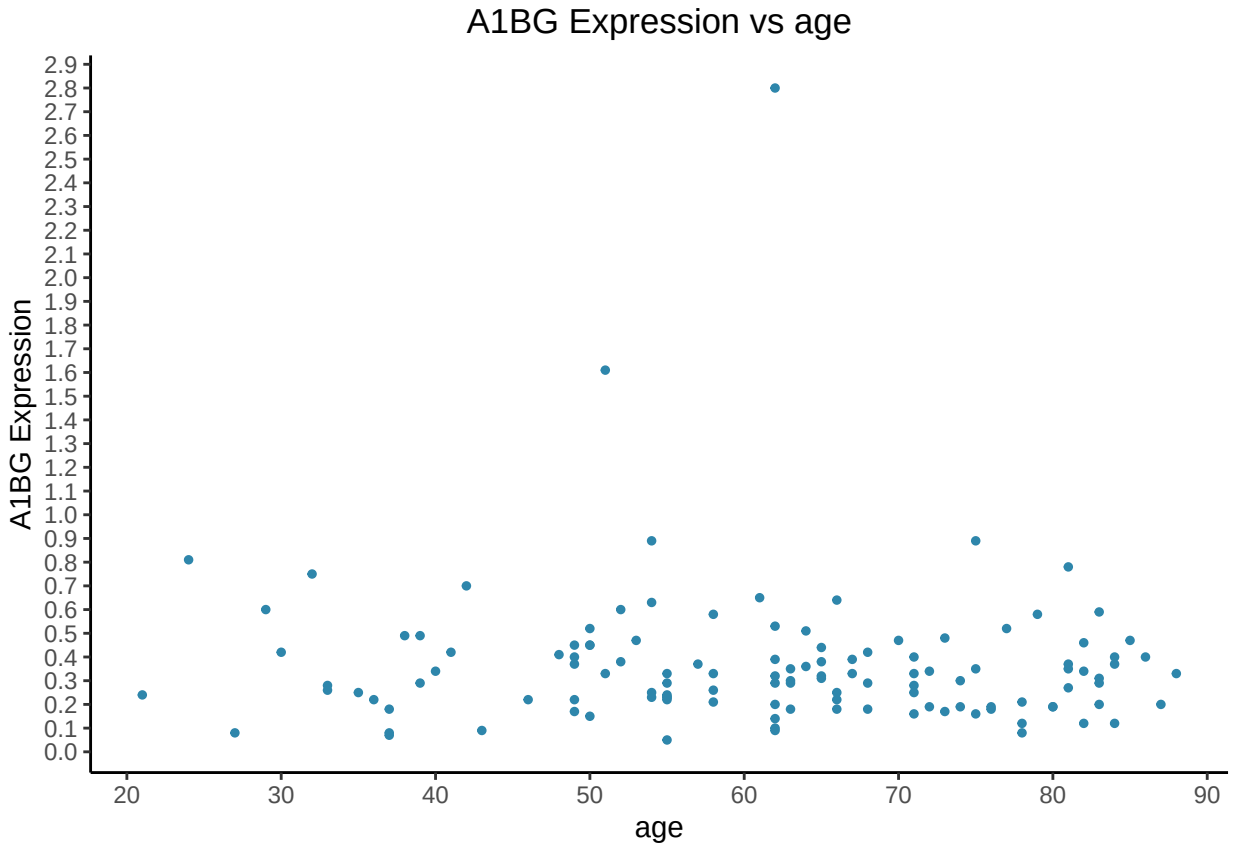
```
library(ggplot2)

# making the function
plot_gene_expression <- function(data, genes, cont_var, cat_var1, cat_var2) {

  # to call function for more than one gene
  for (gene in genes) {

  # histogram
  histogram <- ggplot(data, aes_string(x = gene)) +
    geom_histogram(binwidth = 0.05, fill = "#D1514D", color = "black") +
    labs(title = paste(gene, "Expression"),
       x = paste(gene, "Expression", y = "Count")) +
    theme_classic() +
    theme(plot.title = element_text(hjust = 0.5))

  print(histogram)
```

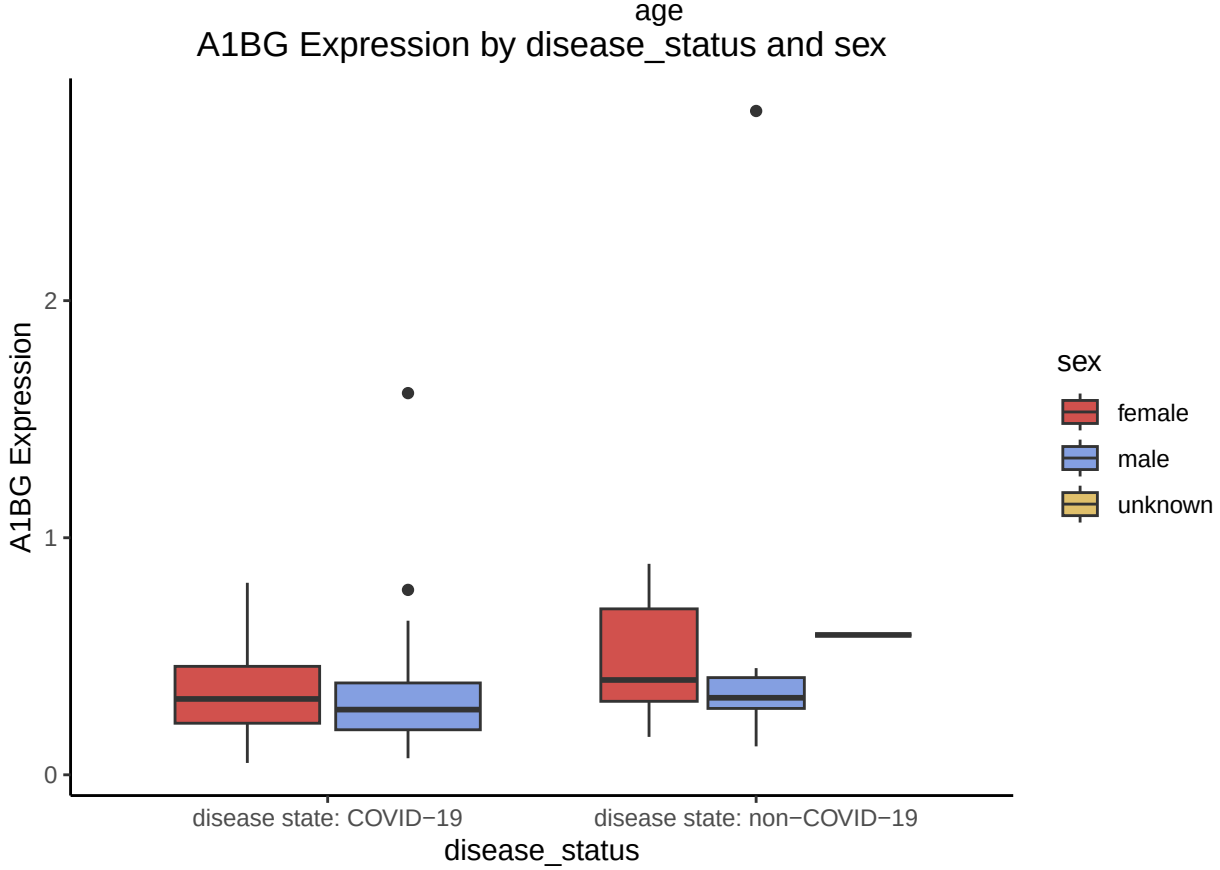
```
  # scatterplot
    # need to make sure the cont variable is numeric
    data[[cont_var]] <- as.numeric(data[[cont_var]])

  scatterplot <- ggplot(data, aes_string(x = cont_var, y = gene)) +
    geom_point(color = "#2E86AB", size = 1) +
    labs(title = paste(gene, "Expression vs", cont_var),
       x = cont_var, y = paste(gene, "Expression")) +
    scale_x_continuous(breaks = seq(0, 100, by = 10)) +
    scale_y_continuous(breaks = seq(0, 3, by = 0.1)) +
    theme_classic() +
    theme(plot.title = element_text(hjust = 0.5))

  print(scatterplot)

  # boxplot
  boxplot <- ggplot(data, aes_string(x = cat_var1, y = gene, fill = cat_var2)) +
    geom_boxplot() +
    scale_fill_manual(values = c("male"= '#849FE1', "female" = '#D1514D',"unknown" = '#E0C06B')) +
    labs(title = paste(gene, "Expression by", cat_var1, "and", cat_var2),
       x = cat_var1, y = paste(gene, "Expression")) +
    theme_classic() +
    theme(plot.title = element_text(hjust = 0.5))

  print(boxplot)
  }

}

plot_gene_expression(data = merged_proj_data,
                     genes = c("A1BG", "A2M"),
                     cont_var = "age",
                     cat_var1 = "disease_status",
                     cat_var2 = "sex")
```

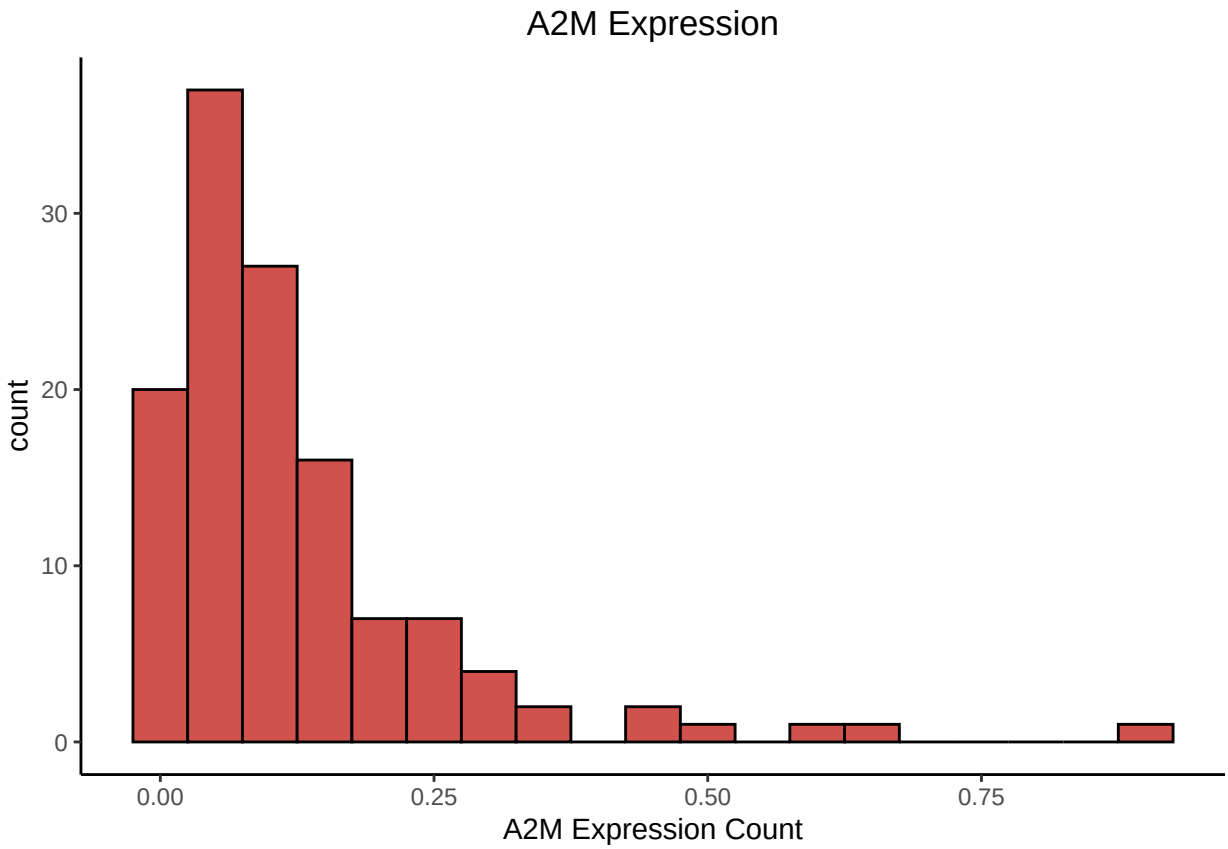
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



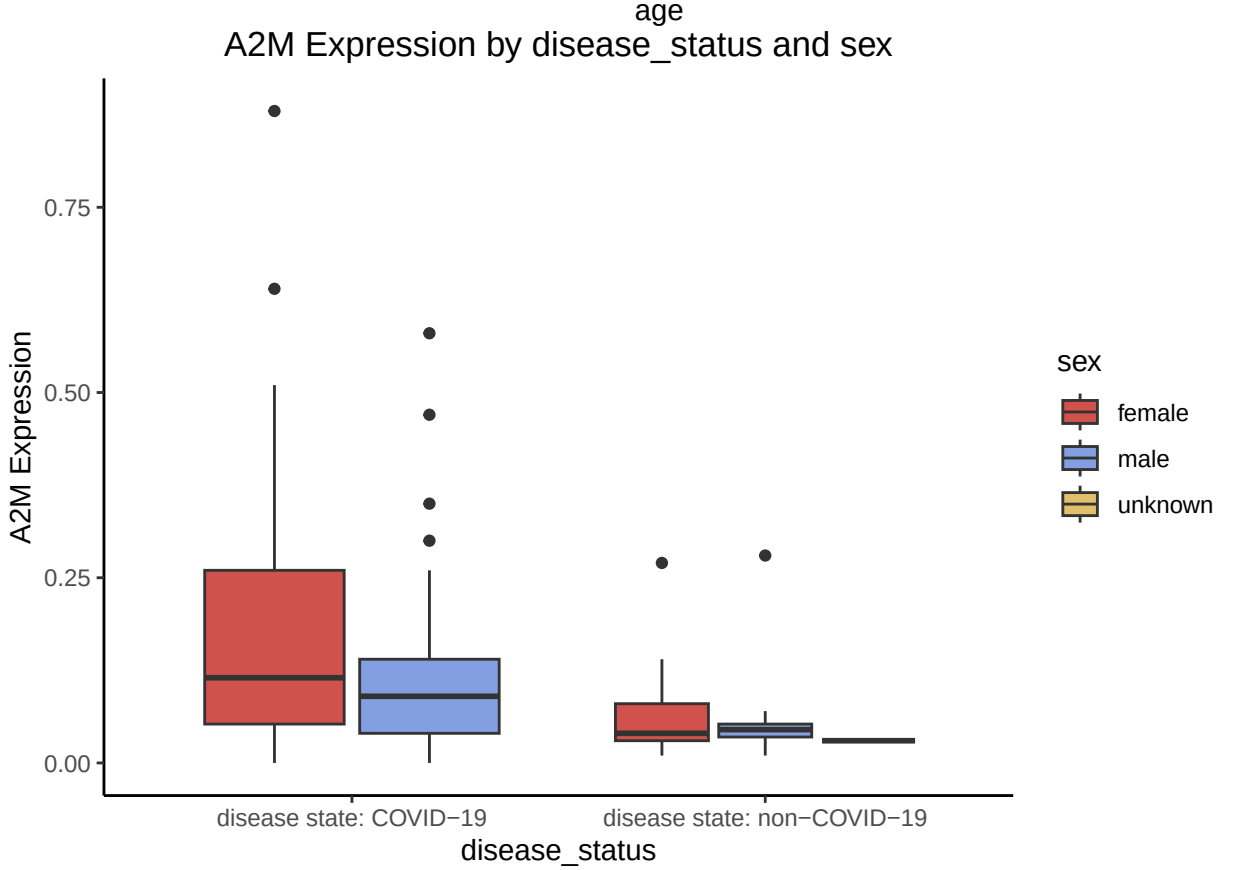

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


A1BG Expression vs age
A1BG Expression
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
2.0
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
20
30
40
50
60
70
80
90
age


A1BG Expression by disease_status and sex
A1BG Expression
2
1
0
disease state: COVID−19
disease state: non−COVID−19
disease_status
sex
female
male
unknown



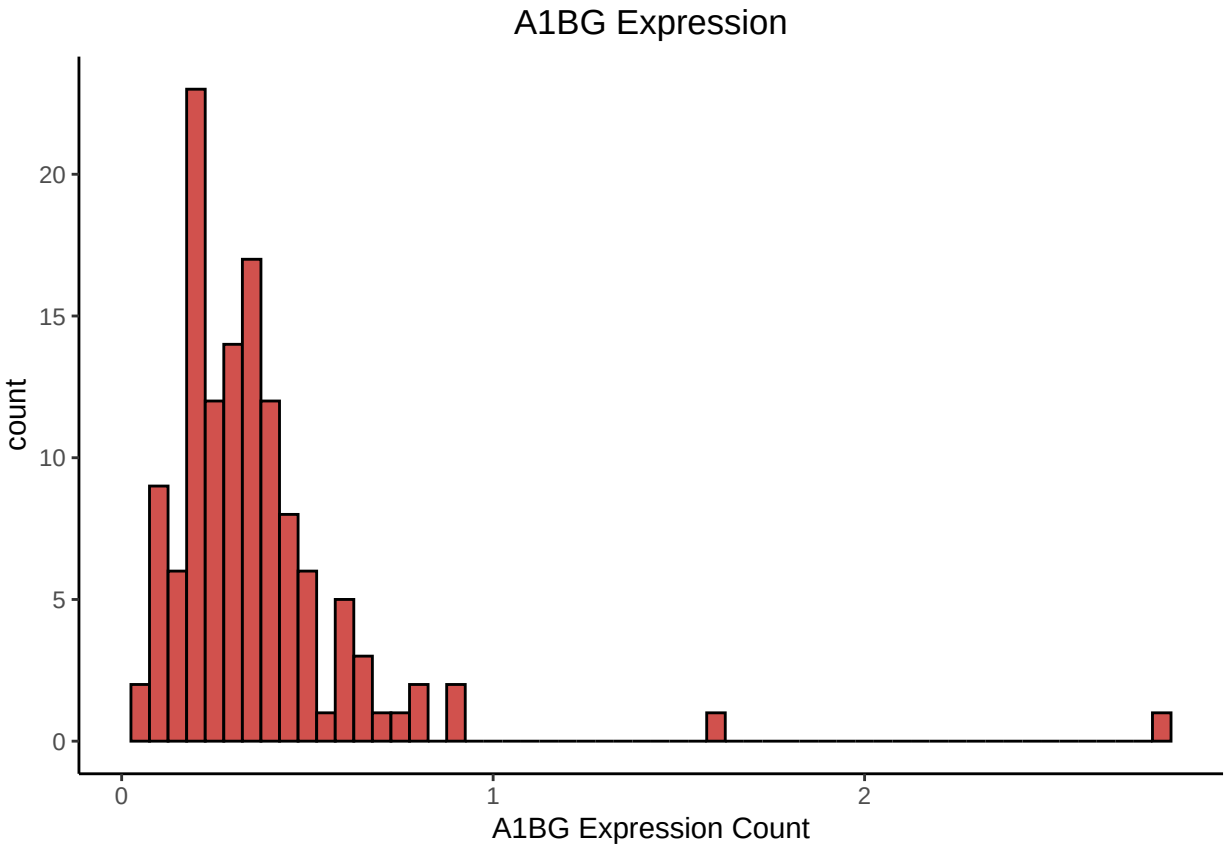

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

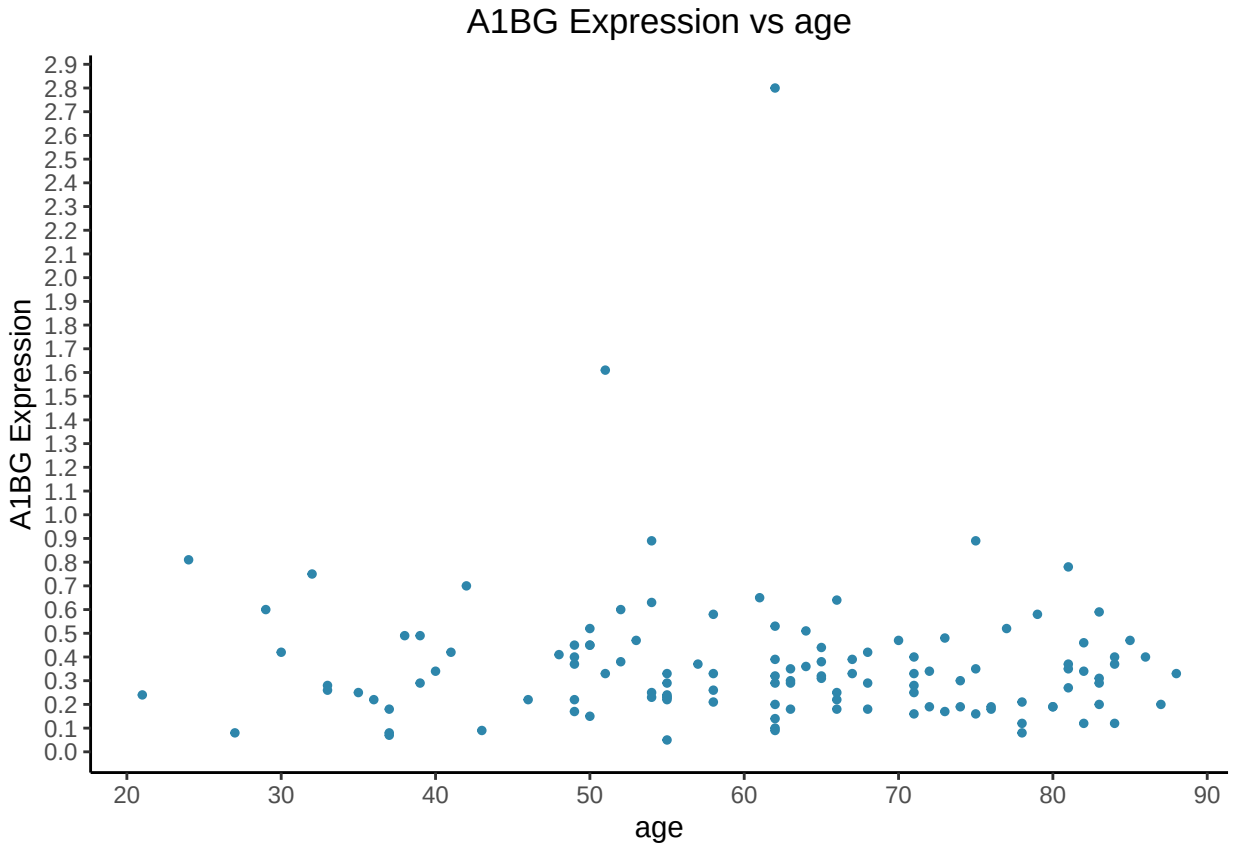

Select 2 additional genes (for a total of 3 genes) to look at and implement a loop to generate your figures

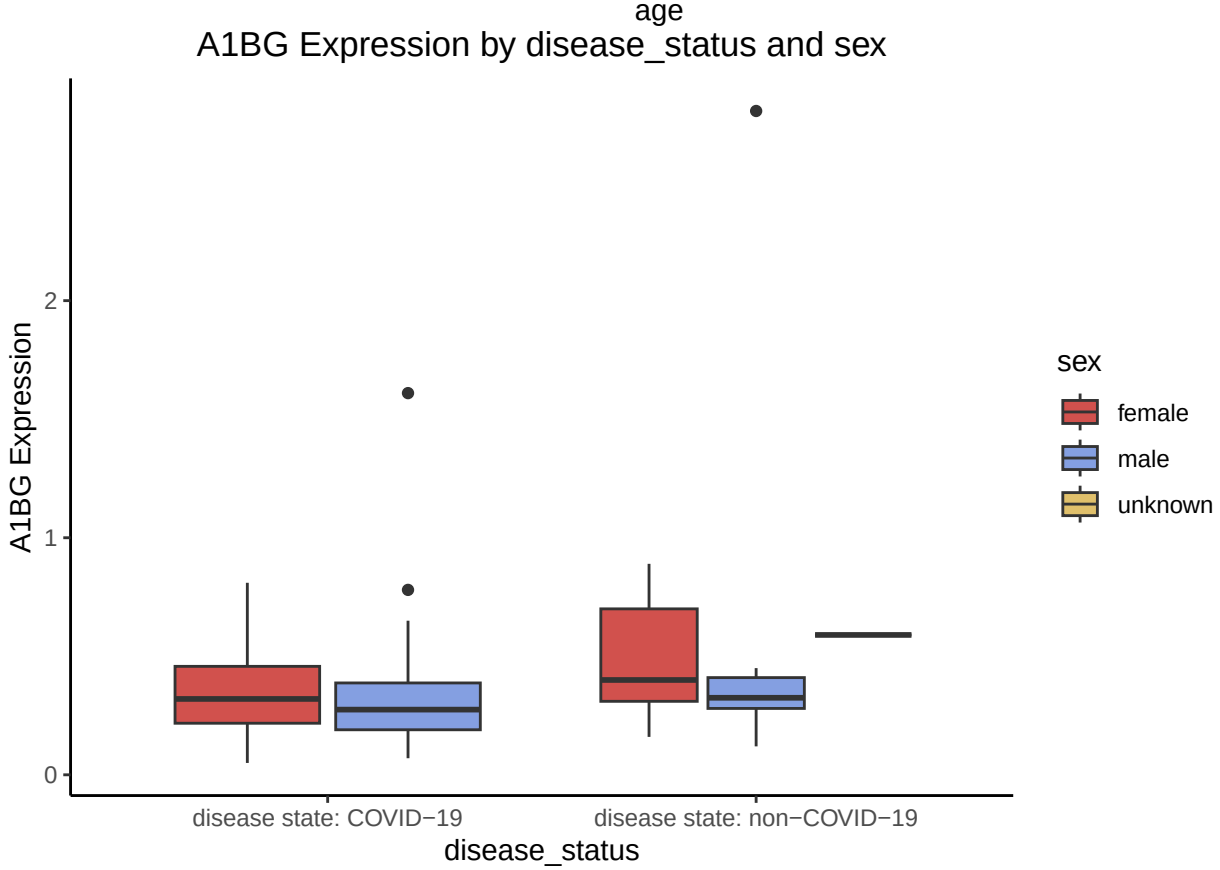
using the function you created (10 pts)

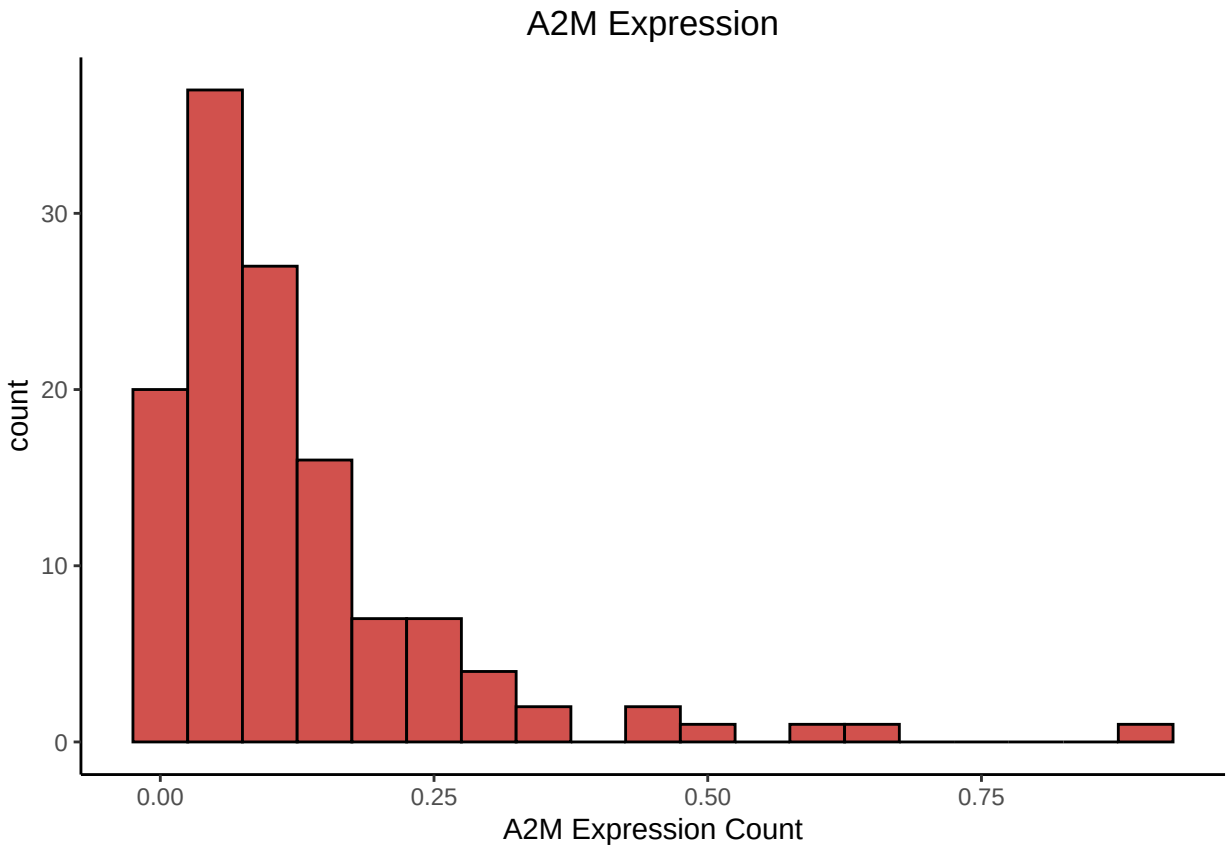
```
new_genes <- c("A1BG", "A2M", "AARD")

plot_gene_expression(data = merged_proj_data,
                     genes = new_genes,
                     cont_var = "age",
                     cat_var1 = "disease_status",
                     cat_var2 = "sex")
```




```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

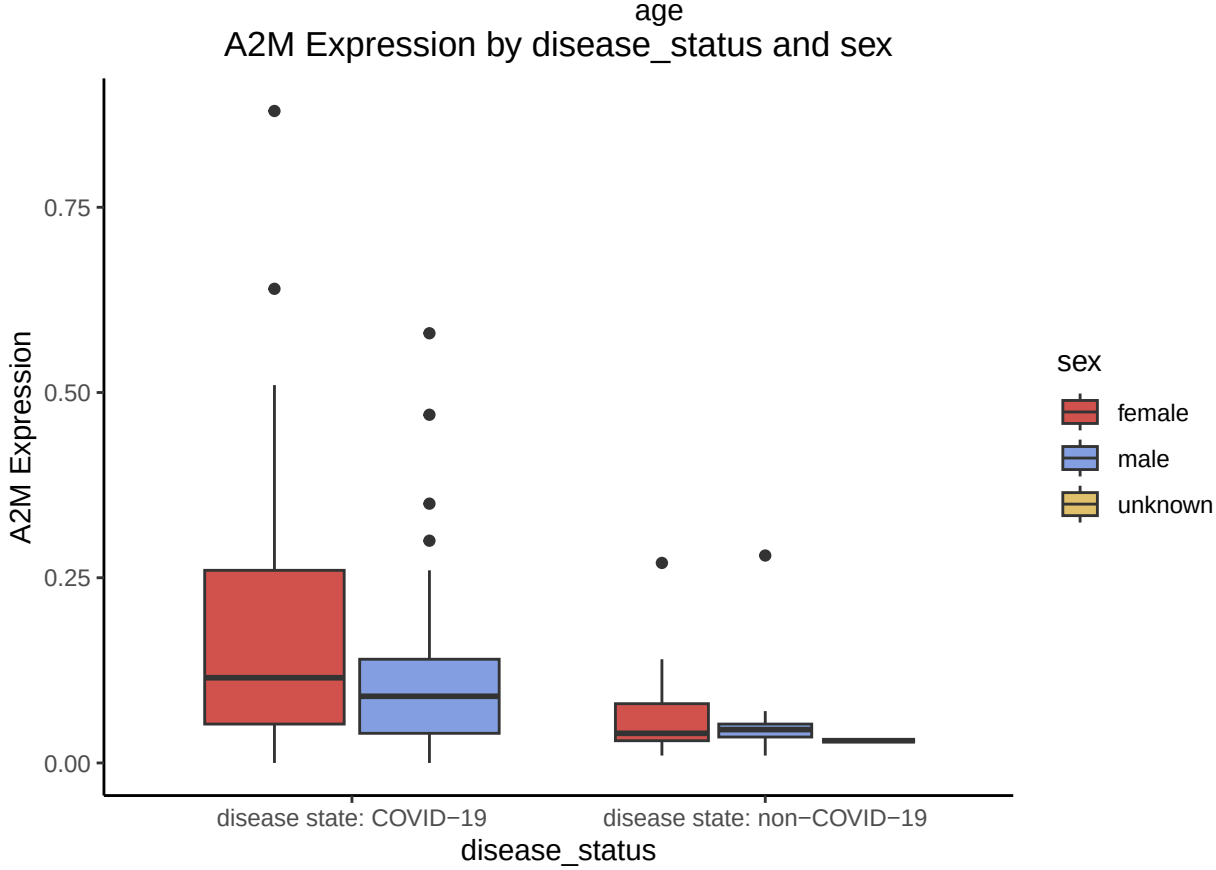

A1BG Expression vs age
A1BG Expression
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
2.0
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
20
30
40
50
60
70
80
90
age

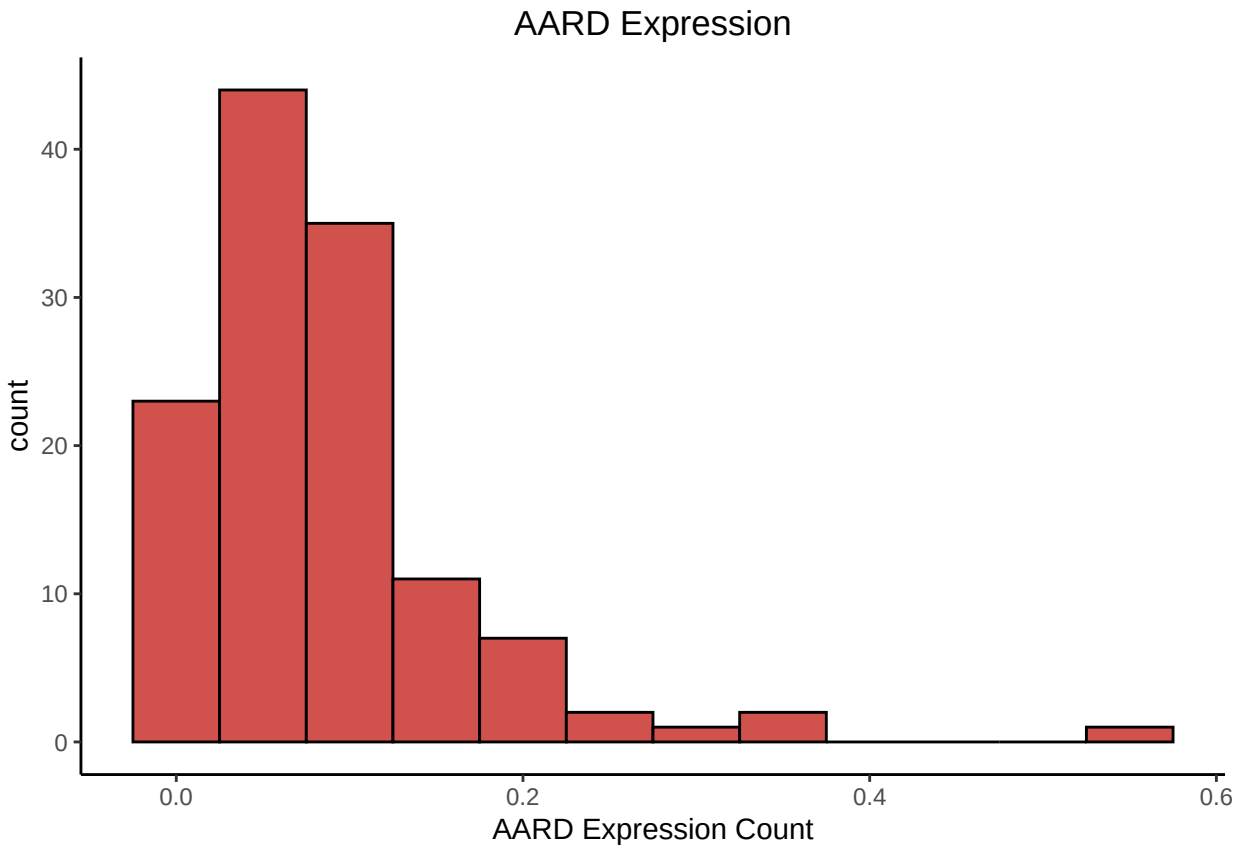

A1BG Expression by disease_status and sex
A1BG Expression
2
1
0
disease state: COVID−19
disease state: non−COVID−19
disease_status
sex
female
male
unknown




```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

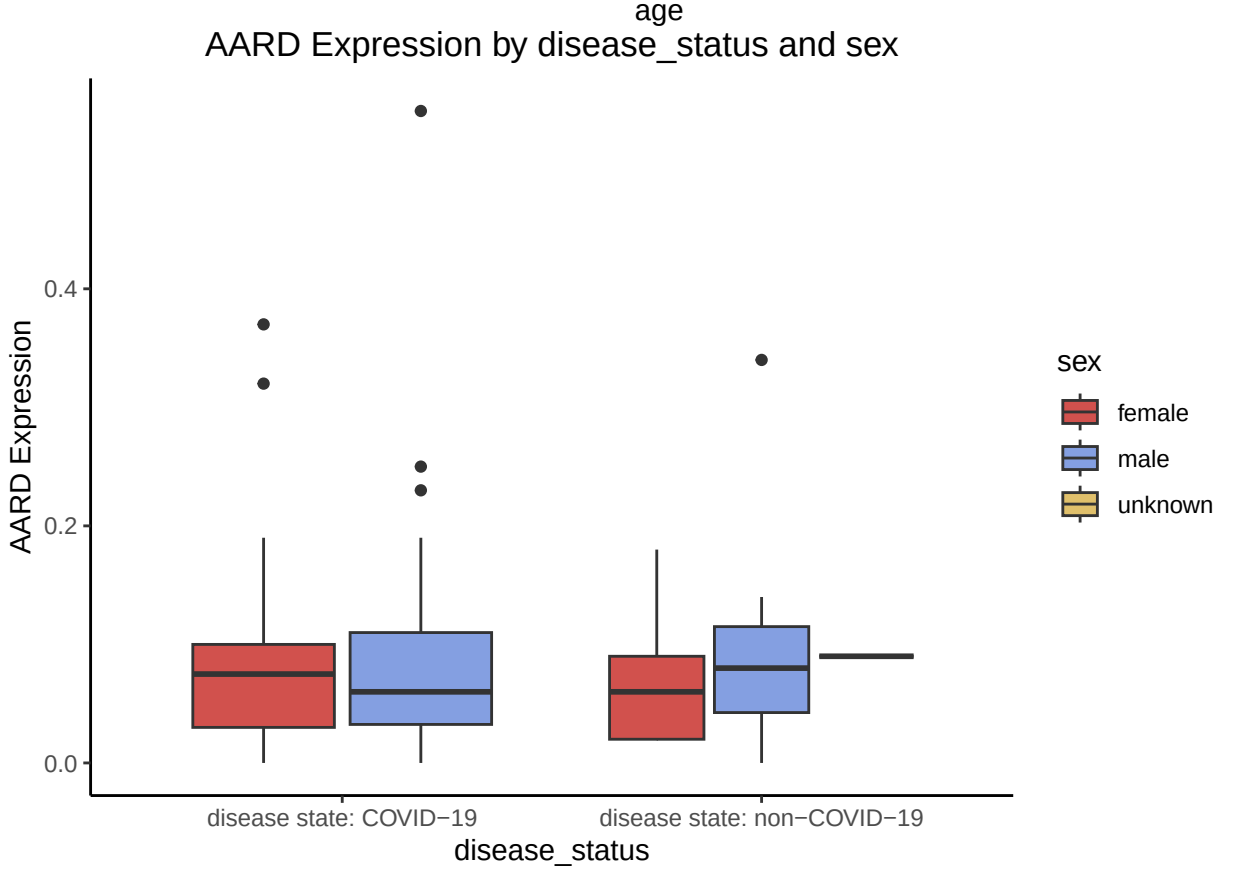

A2M Expression vs age
A2M Expression
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
20
30
40
50
60
70
80
90
age


A2M Expression by disease_status and sex
A2M Expression
0.75
0.50
0.25
0.00
disease state: COVID−19
disease state: non−COVID−19
disease_status
sex
female
male
unknown




```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


AARD Expression vs age
AARD Expression
0.0
0.1
0.2
0.3
0.4
0.5
20
30
40
50
60
70
80
90
age


AARD Expression by disease_status and sex
AARD Expression
0.0
0.2
0.4
disease state: COVID−19
disease state: non−COVID−19
disease_status
sex
female
male
unknown